AUTONOMOUS NAVIGATION ABILITY: FIDO TESTS RESULTS

E. Baumgartner* and M. Maurette**

*Jet Propulsion Laboratory, M/S 82-105, 4800 Oak Grove Drive, Pasadena, CA USA

**CNES 18 Avenue E. Belin 31401 Toulouse cedex France

ABSTRACT

Autonomous navigation of a rover on Mars surface can improve very significantly the daily traverse, particularly when driving away from the lander, in unknown areas. The FIDO platform of the JPL has been used to evaluate the ability of autonomous obstacle avoidance developed by JPL and CNES autonomous long range path planning. The algorithms have been tested in realistic conditions during 2 weeks in January 2000, in the JPL MarsYard. The ability of the rover to reach a distant goal in difficult terrain has been evaluated in several situations. Portability and computing resources evaluation for implementation on a Mars rover have been assessed as well as the maximum daily traverse allowed in autonomous mode. The results show that only a very small amount of energy and computing time is used to implement autonomy and that the capabilities of the rover are fully used, allowing a much longer daily traverse than purely ground-planned strategies. Finally a combination of JPL and CNES navigation has been recommended.

Keywords: rover, stereovision, autonomous navigation, planetary exploration

1. INTRODUCTION

The Mars exploration program settled by NASA and, in particular, the Mars Sample return missions include several rover components that have to fulfill scientific objectives that require long range mobility: the specified daily traverse is 100 meters per day. A pure ground operator based strategy is unable to reach this objective in rough terrain. Autonomous path generation is necessary to execute the daily traverse goal given by ground operators that received panoramic images from the last rover location. This is mainly due to the fact that, except in smooth areas with isolated and scarce obstacles, the ability to plan a safe path from the initial position is limited to about 10 meters maximum distance, and also to the trajectory drift during execution that can lead the rover meters away from the desired path.

To overcome these problems, two navigation strategies have been studied by JPL and CNES, and implemented on experimental rovers. Both are based on stereovision perception, with different implementations, but are quite different in the process leading to on-board path generation. The JPL algorithm uses waypoints provided by ground operators and tries to follow a direct path to the next waypoint. When an obstacle is detected by the vision system and related software, a local path is computed to find a way around the obstacle. The direct path to the next waypoint is then attempted again.

The CNES algorithm [Ref 3] constructs a Digital Terrain Model from the stereo images, then analyses it to determine the navigable areas and to score their difficulty. The result is a navigation map corresponding to a stereo pair, which is then merged with the previously computed ones to get a global navigation map.

Given a distant objective, the algorithm then generates the “best” waypoint inside this map and finds an optimal path to it. Only part of this path is executed and the whole process is restarted to update the path before its end. To evaluate the performances of JPL and CNES vision and navigation software, real tests have been implemented on a rover whose overall architecture is representative of the next generation mars rovers : FIDO. (Figure 1)

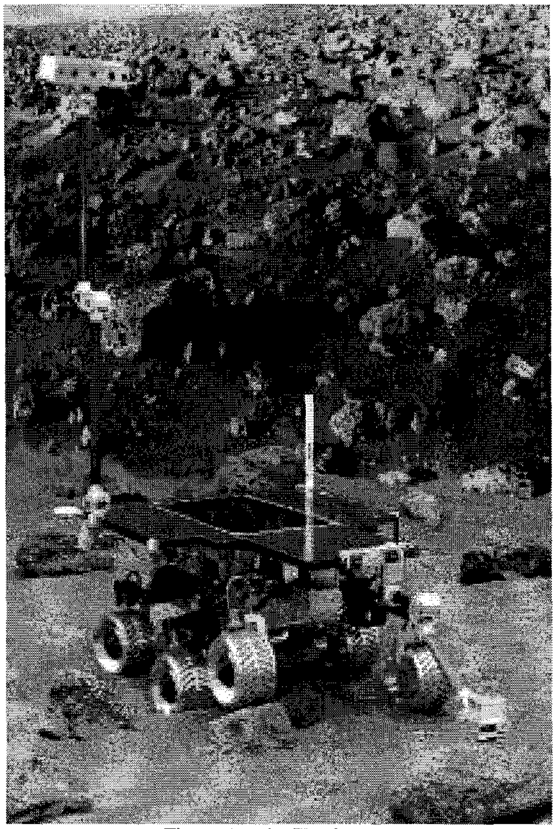

Figure 1 : The FIDO rover

The tests were managed by JPL FIDO team that supplied the vehicle and the test site, implemented the system software and the path execution control, as well as the localization sub-system. The JPL autonomous obstacle detection and avoidance was already implemented on the vehicle for the tests. CNES implemented the vision and navigation software on board and provided the ground station and the display and control software on it.

The ability of the rover to overcome obstacles and reach the goal that has been assigned with a minimal time and power consumption have been addressed placing FIDO in the JPL

MarsYard and comparing the behavior with the different algorithms. The comparison has been made on both vision and navigation algorithms and on the corresponding computation times and memory allocations. The analysis has been done both on single situation results and on complex long-range trajectories.

2. THE FIDO ROVER

The FIDO rover [Ref 1] is composed of a single body locomotion platform [Ref 2] mounted on a rocker-bogie suspension (see Figure 1) that connects to the body via a geared differential through two structural members (Jeff tubes). The 6 powered wheels are metallic and independently steerable. It is equipped with several set of cameras :

- the HAZCAMS used for hazard detection and avoidance are stereo cameras with ultra-wide angle lenses (110° horizontal field of view). They are located on the front and on the rear ends of the body and have limited resolution and range.
- The NAVCAMS are stereo cameras mounted on a four joints deployable mast able to place them at approximately 1.7 m height during static snapshots. Their lenses produce a 43° horizontal field of view on a 512x512 pixels CCD camera.
- The PANCAMS are also located in the box on top of the mast. They are equipped with longer focal lenses and are used by scientists to determine the place and nature of the experiences to be performed

The main body includes the power supply composed of batteries and converters. The solar panel on top of the case can supply additional energy source, but was not used during the tests.

The rover is equipped with odometry sensors which, associated to a sun sensor estimating the heading, provide a relative position estimate used to control the planned motion. The on-board computer is a single board PC104 CPU equipped with a Pentium running at 133 Mhz and its peripherals (among which a frame grabber to acquire the images). The communications with the control ground station are performed using a wireless Ethernet link.

The operating system is VXWORKS and the application software running on board is written in C language. The ground stations used to control operations were a PC and a Sun workstation. The global software architecture is presented in Figure 2

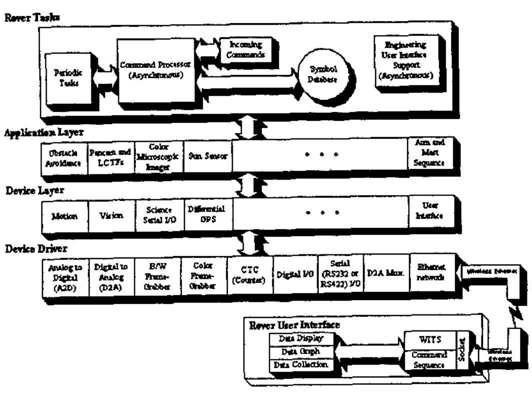

Figure 2 : FIDO Rover software architecture

3. THE TEST SITE

The terrain used to perform the tests was the JPL MarsYard. The obstacle density for the tests was rather high and the size of the stones offered a good range of dimensions so that the classification into obstacles by the on-board software, regarding the climbing capacities of the rover, was tested under varied conditions. The variety of slopes was however more limited on this field. This justified additional tests to be sure that dangerous slopes or combinations of rocks and slopes were correctly detected. A view of the terrain is illustrated in [Figure 3]

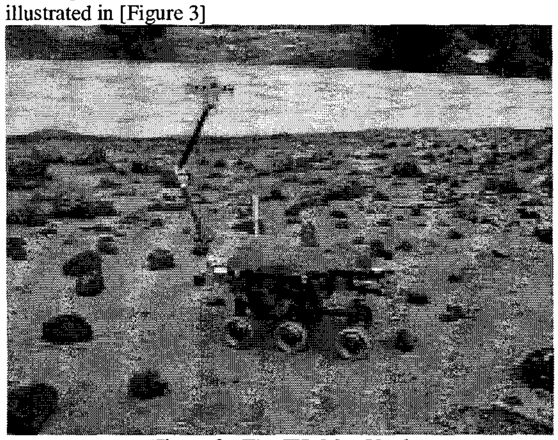

Figure 3 : The JPL MarsYard

4. VISION TESTS

4.1. Tests objectives

The objectives of the tests performed on the vision algorithms were to evaluate the performances and robustness of the algorithms and to measure the necessary CPU resources for different image resolutions.

4.2. Methodology

Vision tests have been performed on an image taken on the Marsyard with the Navcams, used at half resolution and an image taken with CNES stereo cameras at full resolution, and coded on 8 bits. The analysis consisted in disparity maps analysis to compare density and evaluate accuracy of the computed values in some critical areas. It also included computing time measurement on the on-board processor, and memory allocation for data and code. Several weaknesses of the analysis appeared during the tests:

- a valid comparison should be made from the same original images. But the rectification process, which is an important step for the final data accuracy, could not be implemented for both algorithms. This made impossible to compare the NAVCAMS processing at full resolution and the CNES images by the JPL algorithm.
- Precision analysis requires a reference model of the terrain which was not available during the tests, thus limiting the investigation to relative precision inside dedicated zones without attempting to measure the total absolute precision of the 3D model computed by the vision system.

A good preliminary evaluation could however be performed and the portability of the algorithms on a representative on-board computer established.

4.3. Tests results

The filtered disparity maps obtained with FIDO images show that the CNES correlation algorithm produces significantly more dense maps (Figure 4). Although the tests were not sufficient to decide if this results from wrong matching removal in rock-edges areas or not, it appears that in smother areas, the values computed by the CNES algorithm are correct (compared to neighboring cells) and give meaningful additional information.

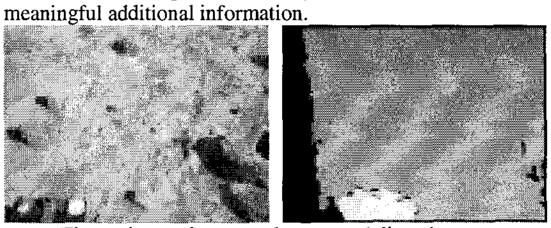

Figure 4 : raw image and computed disparity map

The corresponding Digital Elevation Model (Figure 5) is computed with a high resolution compared to the robot wheel size, thus offering a very good navigability analysis of the terrain.

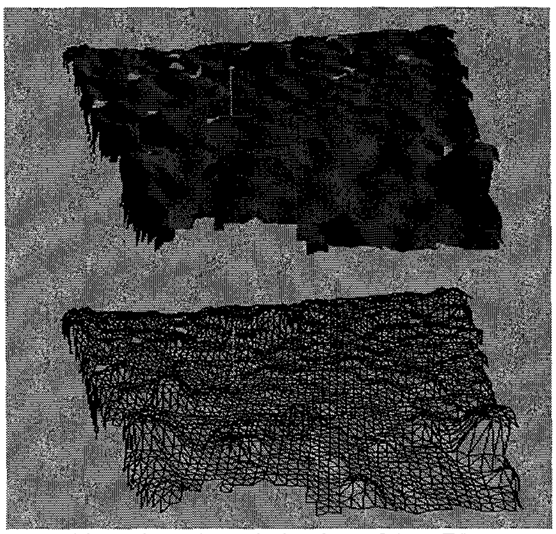

Figure 5 : shaded and wire views of the D.E.M.

A further evaluation will require a reference model of the same area. This will be discussed in the future work paragraph Computing time measurements were performed with several tests sets of images and resolutions and gave the following results. With 486 x 512 pixels Navcams images whose resolution is degraded by a factor of 2, a disparity interval of 4 to 45 and a DEM grid of 50 mm with a total grid size of 251 x 251 cells A total time of 2167 ms was spent by CNES algorithm and 7400 ms by JPL algorithm to reconstruct the DEM. As the total computation time represents less than 5% of the locomotion time, it can be considered that both software can run on board without significant impact on the time and power budgets.

5. OBSTACLE DETECTION

The classification of the terrain according to the climbing capacities of the vehicle have been tested first on simple and easy to check situations: isolated rocks, uniform slopes and combination of both. As a wide variety of slopes were not available on the MarsYard, they have been simulated by setting a bias in the accelerometers information so that the vehicle "feels" the whole terrain tilted. By varying this bias we could verify that a slope was classified dangerous by the image-based algorithm when the maximum allowed slope was reached. Figure 6 illustrates the behavior of the algorithm when the limit is reached: on an almost flat terrain with an average slope just below the maximum allowed, small stones or footprint on the terrain are detected as obstacles as the attitude of the rover will reach excessive values if it was placed on that location.

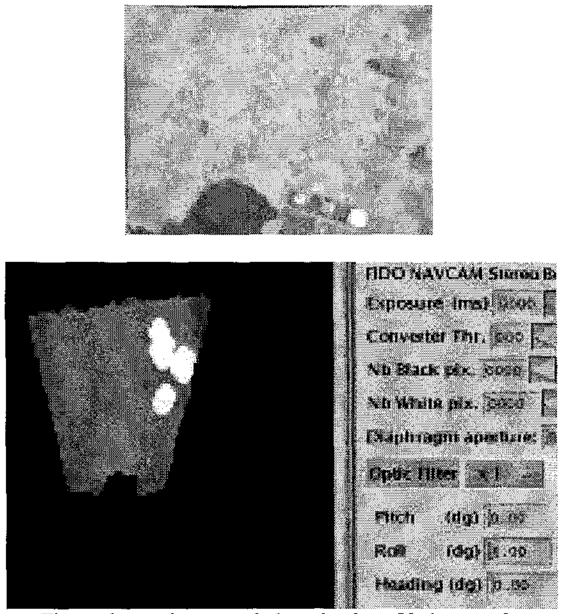

Figure 6 : raw image and obstacles for a 8° slope angle

Increasing the slope angle rapidly leads to a totally non-navigable classified area.

The same approach has been taken for mixed rocks and slopes areas. Figures 7 and 8 show the shape of the non-navigable areas around a rock when the terrain is tilted right or left.

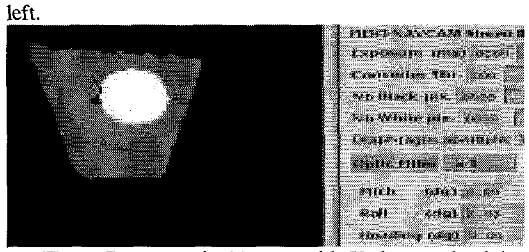

Figure 7 : non navigable area with 5° slope to the right

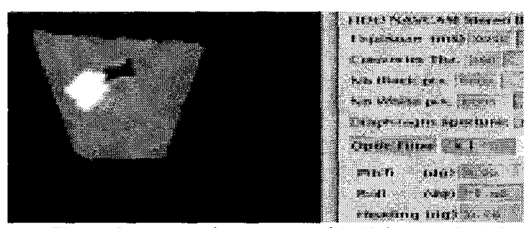

Figure 8 : non-navigable area with 5°slope to the left

The resulting slopes were measured on the real terrain and the validity of the detection has been established.

5.1. Computing time and resources

With 486 x 512 pixels NAVCAMS images whose resolution is degraded by a factor of 2 and a DEM grid of 50 mm with a total grid size of 36 x 32 cells, the following execution time have been measured with JPL algorithm: 216 ms. While CNES algorithm using same images and DEM grid, but with a grid size of 251 x 251 cells was executed in 1533 ms. The total execution time of both algorithms from image acquisition to path generated is thus as follows:

- JPL software: 7616 ms
- CNES software: 3267 ms

In any case the maximum computing time was less than 5% of the locomotion time and is negligible along the mission.

A total memory allocation of 3.0 Mb including the input images is sufficient to run CNES stereovision and navigation software in the conditions of the FIDO tests. This figure is compatible with an implementation on Mars rover on-board computer, as they are foreseen for the next flight opportunities.

6. LONG-RANGE NAVIGATION

6.1. Algorithm description

The results of the both JPL and CNES navigation algorithms used separately have been analyzed executing long traverses across the MarsYard and placing the rover in front of typical difficult situations like a trap-shaped rocks arrangement. To understand the different behavior of the rover when it uses these algorithms, a short description of them can be as follows:

- JPL software: the rover attempts to follow a direct path to the next waypoint given by the ground operator. During the movement, stereo pairs are acquired from the HAZCAMS in front of the rover and obstacle detection is performed from the corresponding D.E.M. When an obstacle is met, heuristics are used to find a way around the obstacle. A low resolution grid (36 x 32 cells) representing the navigability of the terrain around the rover is maintained to keep a short term memory of the strategy used to find a path. When the direct path becomes possible again, the normal progression is restarted.
- CNES software: Only the final goal needs to be given by the ground operator. Then the rover performs panorama acquisition from the start point. The different navigation maps computed from each stereo pair are merged together to obtain a global map (set to 12x12 m around the rover during these tests). This map represents not only the navigable, non-navigable and unknown areas but also the difficulty of the terrain. The algorithm then computes a path that optimizes the progression towards the goal using the easiest areas. Margins corresponding to the localization errors and to inaccuracies in path execution are included. Depending on the terrain characteristics, the path is usually around 5 meters long but only the first half is executed. A new perception is then planned to optimize the knowledge and the resulting navigation map is merged with the global navigation map. Path planning is thus always using a 250 square meter knowledge around the rover.

6.2. Long range tests results

With relatively simple situations (isolated rocks on a smooth surface) and few meters trajectory, both algorithms proved to avoid correctly obstacles and reach the objectives that have been assigned. The interest of a global planning strategy was demonstrated in more complex situations as the two described hereafter.

6.2.1. Multiple obstacles trajectory

The target was set at about 15 meters away in a region were several large rocks had to be avoided on the way. Figure 9 shows a view of the travel to the goal.

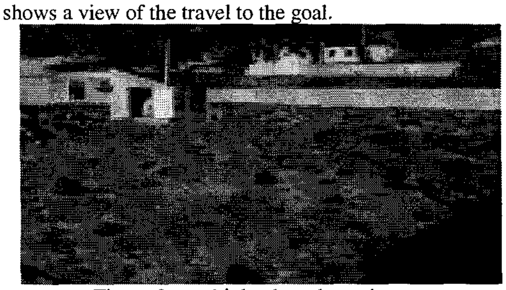

Figure 9 : multiple obstacles trajectory

The rover is on the left side of the terrain and the goal at the extreme right. With the global planning algorithm, the goal has been reached in 6 steps (initial panorama and 6 additional perceptions). The rover used a path that was very near to the shortest way to the goal, using narrow gates between the rocks like in step 3 (Figure 10) or in step 5 (Figure 11).

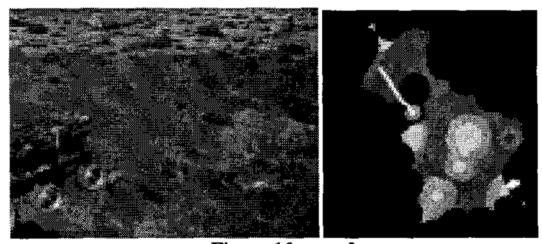

Figure 10: step 3

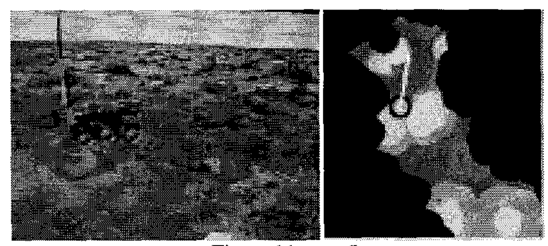

Figure 11: step 5

With the local obstacle avoidance algorithm, the first part of the travel, were the rock density is lower, was rather similar to the previous one, but when arriving near the target, surrounded by large obstacles, the local strategy to avoid the big rock (Figure 12) made the rover miss the shortest way on the left of the rock.

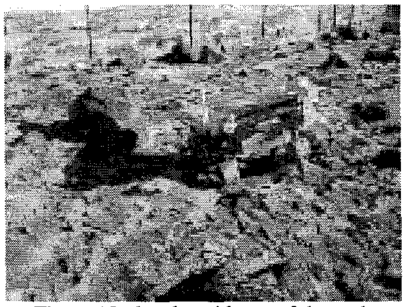

Figure 12 : local avoidance of the rock

The rover had then to follow a longer path before switching again to direct progression as illustrated in the Figure 13 sequence.

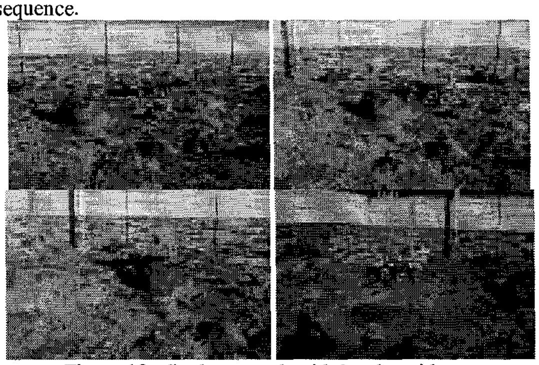

Figure 13 : final approach with local avoidance

6.2.2: "Trap" situation

The "trap" was made with big rocks disposed in a semicircular configuration in front of the rover to create a dead end as illustrated in Figure 14. The goal was set behind the stones line. The more distant rocks were far enough to be out of vision reach of the initial perception of the rover to make it enter into the trap.

Figure 14 : dead end test

With the obstacle avoidance algorithm, the rocks were identified as obstacles and a right escape strategy was initiated. Successive perceptions and obstacle avoidance on the right side led the rover up to the entrance of the trap as depicted on Figure 15. At that time, the further obstacles were far enough to be out of range of the local planning. The rover could not then avoid entering the trap again when trying to reach the waypoint, resulting in failing to find a path. Although the obstacle detection worked well and generated avoidance trajectory consistent with the obstacle's position, the planing range was too short to get out of the trap.

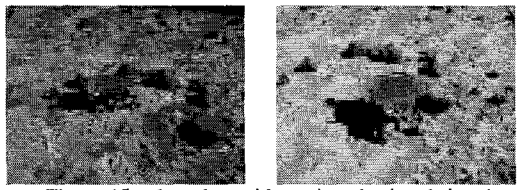

Figure 15 : obstacle avoidance in a dead-end situation

With the global planning algorithm, starting with the initial panorama (5 images) 6 perceptions were performed by the rover. It first attempted to escape the trap on the left side. After a new perception, the navigation algorithm identified that the left side was closed and it attempted to escape by the right side. The rover could find a way around the right stones. (Figure 16)

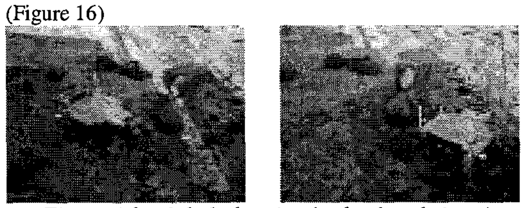

Figure 16 : global planning in dead-end situation

It is interesting to note that, after the way was found; progression was executed in several steps because the escape trajectory was found using already old perceptions. In that case, for safety reasons related to possible drifts during path execution, the navigation algorithm requires a new perception before executing the path in order to refresh the obstacle's position. The first image of Figure 17 shows that a navigable area around the rock has been identified (in gray) but a path is planned only on the next (right image) map.

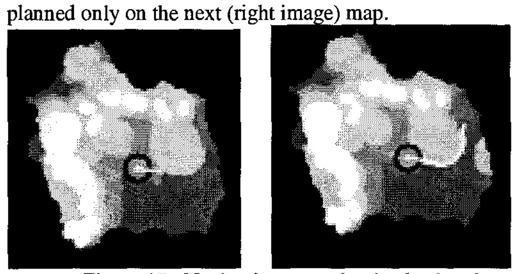

Figure 17 : Navigation maps for the dead-end

7. CONCLUSIONS AN RECOMMANDATIONS :

The tests have shown that global planning navigation algorithms give a significant improvement to the rover daily traverse ability when crossing difficult areas. The required computing time and resources remain low compared to other budgets, and the portability of the software to an on-board computer has been established.

An efficient implementation of the navigation will be to have CNES navigation running at the higher level, using the NAVCAMS, and obstacle detection, using HAZCAMS running during path execution to guarantee an optimum safety with two independent (hardware and software) subsystems contributing to the rover autonomous navigation. This will also present the advantage that the navigation of the rover remains possible in case of failure of either NAVCAMS (switch to obstacle avoidance strategy) or HAZCAMS (switch to unmonitored CNES navigation) without adding any hardware. This approach has been taken as the baseline for future work. A second test period is foreseen in 2001 with integrated on-board software and off-line monitoring on ground to reproduce a realistic situation for Mars exploration.

ACKNOWLEDGEMENTS

The research associated with the FIDO rover described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

REFERENCES

[Ref 1] Baumgartner E. "Exploration Technology Rover." http://robotics.jpl.nasa.gov/tasks/etrover/homepage.html

[Ref 2] Lindemann R., Reid L., Voorhees C. "Mobility Sub-System for the Exploration Technology Rover."

[Ref 3] Rastel L., Delpech M. "Enhanced path planning and localization techniques for autonomous planetary rovers" ASTRA 98 Workshop.